GALIL
WE MOVE THE WORLD

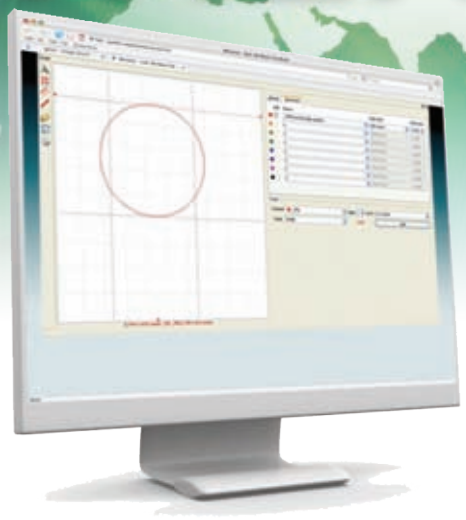
SERVO TRENDS

APRIL 2009, VOL. 24 NO. 2

QUARTERLY NEWSLETTER PUBLISHED BY GALIL MOTION CONTROL

"The mission of Galil's experienced Applications Department is to provide prompt and accurate technical assistance to help OEMs successfully deliver their products to market."

GALIL SUPPORT TEAM



GalilTools Scope now plots XY circular paths

Galil Releases GalilTools Version 1.3.0.0

By Jason Rhodewalt, Applications Engineer

GalilTools is a software package that allows easy set-up and analysis of Galil control systems. The full version of GalilTools includes helpful tools including:

1. Scope tool for looking at real-time data such as position and speed.
2. Tuner tool for automatically adjusting PID servo parameters.
3. Watch tool for displaying controller status such as I/O and motion.
4. Terminal tool for sending and receiving Galil commands.
5. Program Editor tool for writing controller programs.
6. Communication Library for use with programming languages such as VB, C#, C++, LabView™ and more.

The latest release of GalilTools, version 1.3.0.0, provides many new features including the ability to save projects, an enhanced scope, an enhanced watch window, and support for Ethernet interrupts.

The new Projects feature allows the user to save the current workspace. Upon opening a new instance of GalilTools, the software will open the program, watch window, and scope with the same settings as defined in the previously saved project file. By saving project files for specific tasks, the user can save set up time for specific watch window and/or scope settings. Some common projects might be a project for tuning, a project for troubleshooting, and a project for developing.

The scope tool in the full version of GalilTools has been packed with many new features. Narrowing in on the data you want to see becomes easier with the new auto-zoom feature or with the recently added click-and-drag feature. Measurement of the data is now easier with the addition of cursors which are used just like a standard oscilloscope. Performing calculations on the data captured in the scope is simpler with the addition of the

➤(cont. pg 2)

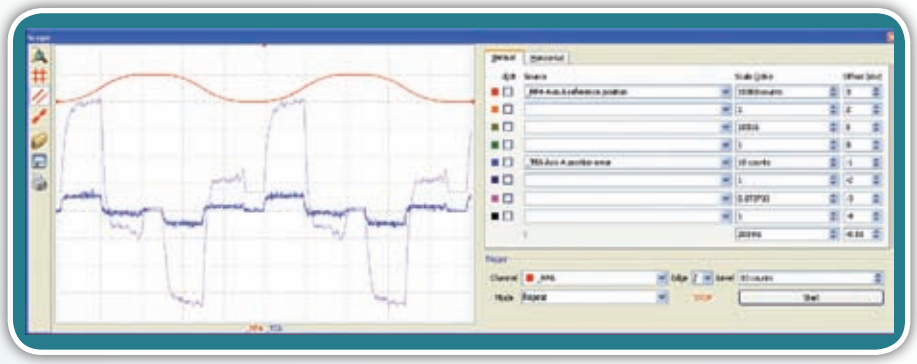
Continued from page 1

import and export options. The user can export data captured in the scope to a .CSV file and use Microsoft Excel to manipulate it. Comparing current traces to past traces can be accomplished with the new scope overlay feature. This can be particularly helpful when tuning a system as it allows the user to see the performance of a move before and after adjusting a tuning parameter, as shown in ***Figure 1***.

The final feature added to the scope is the ability to change the source information for the horizontal axis. Time is still the default for the horizontal axis, but now the user can change the source to any data available in the scope. This is useful for many reasons, including graphing vector motion. By graphing the A axis encoder position on the X axis of the scope and the B axis encoder position on the Y axis we can graph all vector motion, including circles. ***(See Figure 2)***.

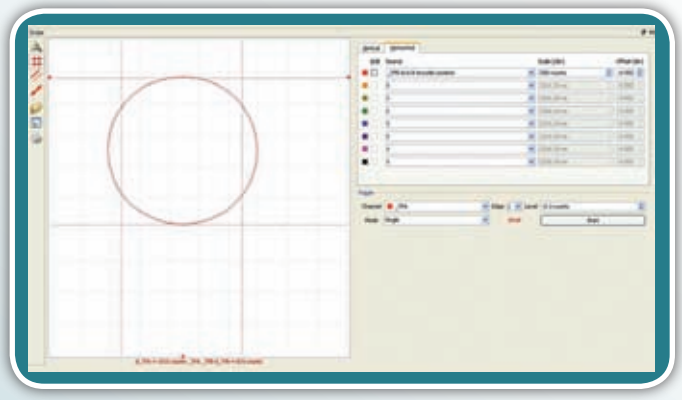
Enhancements have been added to the watch window as well. These include the ability to rename data points along with scaling and adding an offset to the value. An example of using these new features would be to create a watch window with descriptions and units your repair technicians can understand as shown in ***Figure 3***. Support for Ethernet interrupts has also been added to the latest GalilTools release. More information about this feature can be found in the "New Features in the Accelera Command Set" article in this issue of ServoTrends. ■

Figure 1: Using the overlay feature the old capture is shown in lighter colors and the new capture in bold colors. In this example changing the PID parameter's significantly lowered the position error.



For further information on GalilTools version 1.3.0.0 enhancements please see the GalilTools product page at http://www.galilmc.com/products/galiltools.php or contact a Galil Applications Engineer at 1-800-377-6329.

Figure 2: Graphing _RPB vs _RPA allows the user to view 2D moves such as the circle shown.



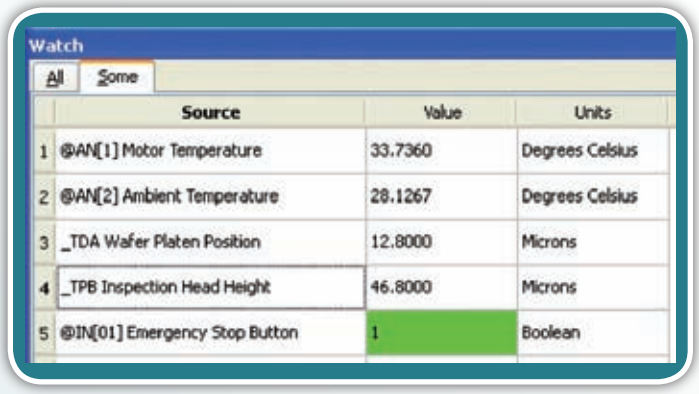


Figure 3: The new watch window allows the user to change the description and user units to something specific to their system.

GALIL
WE MOVE THE WORLD

DMC-40x0 and RIO Now Available with ETL and CE Certification

Galil's DMC-40x0 Accelera motion controller series and RIO-471xx Pocket PLC series are now available with ETL certification. ETL certification is a standard, no-charge feature for the RIO series. There is a $50 adder for the DMC-40x0 with ETL. To order a DMC-40x0 with ETL, specify –ETL at the end of the part number. Both the RIO-471xx and DMC-40x0 also come standard with CE certification.

***ETL** certification indicates that the product has been tested to and has met the minimum requirements of a widely recognized U.S. product safety standard. The ETL Mark is accepted throughout the United States when denoting compliance with nationally recognized standards such as ANSI, IEC, and UL.*

***CE** marking is a mandatory European marking for certain product groups to indicate conformity with the essential health and safety requirements set out in European Directives.*

RIO-47200 Pocket PLC with Screw Terminals Now Available with Metal Cover

In January, Galil announced the RIO-47200 with screw terminals. The RIO-47200 is now available with a metal cover as standard at no additional cost. The metal cover offers additional protection in industrial environments.

Galil's RIO Pocket PLC series is smart, compact, low cost and packed with I/O. Standard features include analog and digital I/O, Ethernet and RS232 ports, a RISC processor for fast I/O handling, PID process control loops, counters, timers, data logging, ability to send email alerts, Power-over-Ethernet, and more. Use the RIO stand-alone or network multiple RIOs to handle all the I/O in your machine. There are several RIO options available as shown in ***Table 1***.

Visit http://www.galilmc.com/products/rio-47xxx.php to download complete specifications and view our new 2 minute video. ■

RIO-47200 with metal cover

Table 1: RIO Options

	# Digital inputs/outputs	# Analog inputs/outputs	Analog resolution	Memory	Connectors & Dimensions
RIO-47100	16/16, 8 outputs are high power	8/8 (0-5V)	12-bits	200 lines, 126 var, 400 array	d-type 3.88x 4.26 x1.3"
RIO-47120	16/16, 8 outputs are high power	8/8 (0-5V,0-10,+/-10,+/-5)	12-bits std, 16-bits opt.	200 lines, 126 var, 400 array	d-type 3.88x 4.26 x1.3"
RIO-47102	16/16, 8 outputs are high power	8/8 (0-5V)	12-bits	400 lines, 254 var, 1000 array	d-type 3.88x 4.26 x1.3"
RIO-47122	16/16, 8 outputs are high power	8/8 (0-5V,0-10,+/-10,+/-5)	12-bits std, 16-bits opt.	400 lines, 254 var, 1000 array	d-type 3.88x 4.26 x1.3"
RIO-47200	16/16, all outputs are high power	8 inputs (0-5V) (analog outputs and 0-10, +/-10,+/-5V optional)	12-bits std, 16-bits opt.	200 lines, 126 var, 400 array	Screw terminals 7.19 x 3.52"

GALIL
WE MOVE THE WORLD

New Features in the Accelera Command Set

By DJ Roberts, Applications Engineer

The Accelera motion control family is Galil's latest generation of ultra-high-speed controllers. Now the Accelera series is even more powerful with the release of new firmware (DMC-40x0 firmware 1.0b and DMC-18x6 firmware 1.0c). This release introduces two powerful new controller features:

1. The concept of an interrupt, as found on PCI controllers, has been ported to the DMC-40x0 Ethernet controller. This has provided a level of compatibility and simplicity in host programming which was until now more complicated and non-standardized.

2. The already powerful multithreaded program stack has been enhanced to allow values to be passed to and returned from stack calls.

These two features have further enhanced Galil's flagship Accelera line.

1. Interrupts: PCI and now Ethernet

A host PC in charge of managing an application has a lot of tasks to balance. Motion supervision is just one in a large list of possible chores. From data analysis, PLC management, computer vision, safety monitoring, user interface, and data logging tasks, the more autonomous the motion control aspects can be, the better. All Galil controllers support a full-featured firmware and an embedded program space that allows the digital motion controller (DMC) to run independently on Galil's embedded real-time operating system.

This autonomous operation is at the foundation of stand alone Galil applications which do not employ a PC. But what about the applications where a PC is central to the management of the application? When a PC is responsible for machine management, the communication between a PC and the DMC is critical.

As an example, consider an application where a frame grabber provides image data to a computer vision application. The PC uses image analysis algorithms to derive position data which is then dispatched to the motion controller. Galil controllers contain motion buffers which allow position data to be pushed to the controller by the host and then used when needed by the DMC. As with any buffer application, the buffer needs to be filled by the PC when needed to avoid a buffer under-run condition. Sending data too soon, however, can result in a buffer over-run condition. How then should the PC decide when to send more data? There are two general methods:

—Polling—

Galil provides buffer operands which indicate how much space remains in the various motion buffers. The PC can poll for this data in a loop, waiting for the available number to be sufficiently large to indicate a new burst of data. Polling is an inefficient method for determining a change in state, however, as each polled value which does not yield a decision to proceed is unnecessary overhead on the PC's microprocessor, the communication bus, and the DMC's microprocessor ***(Figure 1)***.

DMC-40x0 Ethernet Accelera Controller Now Has Communication Interrupts.

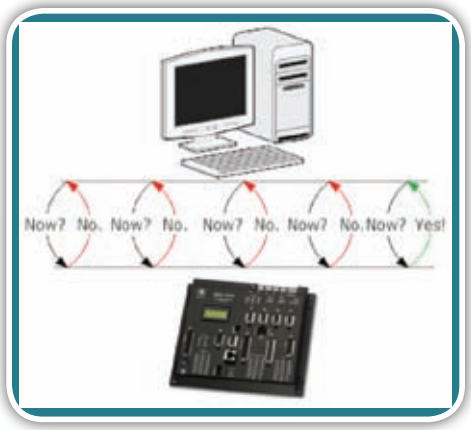


Figure 1: Polling is very inefficient. In this example, only 1 in 5 polls provides new information.

—Interrupts—

An efficient, high-performance method of communication uses the concept of an interrupt signal to tell the PC when to send more data. Using this method is 100% efficient, in that no overhead communication is required to maintain the buffering ***(Figure 2)***.

Figure 2: Interrupts release the host and the DMC from servicing polling.

GALIL
WE MOVE THE WORLD

Continued from page 4

Interrupts are a fundamental hardware feature on the PCI bus. Galil's PCI controllers, such as the DMC-18x6, have always had interrupt support. By specifying which interrupts should be generated with the EI command, the controller will automatically interrupt the host PC when the specific event occurs. User interrupts (UI) are also provided to allow DMC code to trigger interrupts directly from embedded code.

Revision 1.0b of the DMC-4000 firmware has ported this interrupt functionality to Ethernet. The interrupt signal is a UDP packet dispatched on the event instead of a discrete digital output on the PCI bus. This difference is abstracted away from both the DMC programmer and the Host programmer with identical syntax from DMC code and identical function calls in the GalilTools programming API, interrupt(). http://www.galilmc.com/support/manuals/galiltools/library.html#interrupt

In addition to 16 unique user interrupts callable directly from embedded DMC code, the firmware can be configured to automatically generate interrupts on the following events as shown in ***Code 1***.

Code 1: Interrupt Events, excluding user definable Interrupts (UI)

```
Axis A profiled motion complete
Axis B profiled motion complete
Axis C profiled motion complete
Axis D profiled motion complete
Axis E profiled motion complete
Axis F profiled motion complete
Axis G profiled motion complete
Axis H profiled motion complete
All axes profiled motion complete
Excess position error
Limit switch Event
Watchdog Timer (PCI only)
Application program stopped
Host command done (PCI only)
Digital input(s) 1-8 active
```

Interrupt Example, X Axis Motion Complete

A very common procedure is for the host PC to command a move, sleep until the move is completed, during which other threads can fully utilize the processor(s), and then command another move once the DMC is complete. The example shown in ***Code 2*** is a C++ example demonstrating this using either Ethernet or PCI interrupts to optimize the communication overhead while still delivering optimal performance.

Code 2: C++ example snippet of interrupt driven synchronization between DMC and host

```
Galil g("192.168.1.12 -MG 0 -DR 0");//connect to DMC-4000 1.0b or better
//Galil g("galilpci1");//connect to PCI, 18x6,18x2
cout << g.connection() << endl;//print connection information
g.command("EI1");//turn on interrupts for "X axis motion complete" (see EI in command reference)
int i = 0;//counter
while( i < 100){
  g.command("PRX=1000;BGX");//profile motion forward
  while(g.interrupt(-1) != 208);//g.interrupt(-1) sleeps this thread until an interrupt is received.  208 means X axis complete
  g.command("PRX=-1000;BGX");//profile motion backward
  while(g.interrupt(-1) != 208);//no communication or processor overhead is consumed while waiting for 208 interrupt
  i++;
}
```

2. The Program Stack: How Functions help DMC programming

The Galil programming language, while simple and easy to learn, provides a very powerful interface to the underlying hardware. Features and abstraction techniques such as variable axes, multi-threaded program design, automatic subroutines (similar to Events in Microsoft languages), and a program stack have made DMC code a formidable tool in motion application development, often holding its own to general purpose, higher-level languages. Now a new feature added to the DMC program stack has further increased the power of Galil's programming language.

To understand the concept of a program stack, consider the following analogy. Imagine you're making a salad. You chop lettuce, carrots, onions, olives, and so on and mix them into a large bowl. Now you need to make the dressing; taking a separate bowl and mixing olive oil, balsamic vinegar, and salt and pepper together to form vinaigrette. Only after the dressing is mixed, is it ready to be served with the salad. A separate bowl was used to create the dressing, and then the results of that bowl were returned to the salad. If your friend didn't like vinaigrette, you might use another bowl to mix up a ranch dressing.

A program stack works in a similar fashion. The main code (the salad) can call other code (the dressing bowl) to be run. When that code is finished (dressing complete), the stacked code returns program execution back to the main code (salad preparation continues). Often times information can be sent from the main code to the stack code to allow it to know how to proceed (e.g. make vinaigrette or ranch dressing), and the results of the stack code can be returned to the main code when ready (finished dressing). Like with various mixing bowls, the same general algorithm (mixing dressings), can be customized with passed values (ranch or vinaigrette), and can even be run simultaneously in multiple threads (extra chefs).

Galil has traditionally had a program stack which provided for calling code from the main code, and returning to the main code after completion. This was accomplished using the jump to subroutine command (JS). The newest Accelera firmware takes JS one step further by allowing the main code to pass parameters to the program stack, and allowing the stack to pass a return value back. This allows for powerful code reuse and code abstraction techniques in DMC code.

To start with a simple example, consider the case where multiple arrays are constantly being dimensioned and deallocated in a dynamic data processing application. When a Galil array is dimensioned, the array values are not zeroed. This means that stale data remains in the array memory even after old arrays are deallocated, similar to C data. A common procedure when allocating arrays would be to zero the array elements before use. This would ensure that no stale data is present while working with arrays.

In ***Code 3***, the stack is used to make this a very easy and efficient operation. The array is passed to the function, along with the index of where to start zeroing. The function then zeros the array from the starting index to the end, and then returns the last array zeroed, effectively the length of the array. This function is general purpose as any size array can be passed, with any array name.

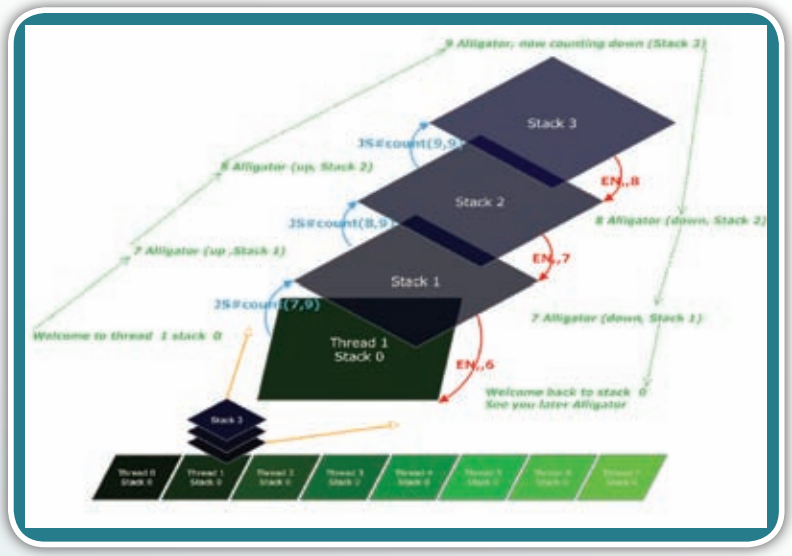
Code 3: Simple array zeroing example using the stack.

```
DMdata[123];'dimension an array with 123 elements
JS#zeroAry("data",0);'zero out buckets starting at index 0
MG"zeroAry zeroed through index",_JS;'print returned value
EN
'
#zeroAry;'(^a arrayName by ref, ^b start by value) zeros array starting at index ^b
^a[^b]=0;'zero element ^b in array ^a
^b=^b+1;'increment element ^b
JP#zeroAry,(^b < ^a[-1]);'[-1] returns the length of an array
'loops until ^b = the array length. JP does not push another stack
EN,,^b;'return the final index to the calling stack
'TERMINAL OUTPUT:
' zeroAry zeroed through index 123.0000
```

To highlight some of the powerful subtleties of the new stack, consider ***Figure 3***, in which the program stack is used to count up the numbers between a starting number, a, and an ending number b, and then count back down from b to a. The counting algorithm used by each stack is always the same, shown in ***Code 4***. Each stack examines the values of a and b passed to it. If the values are equal, the counting up is complete, and it simply returns b-1 to the stack below it to begin the counting down procedure. If a<b, then the stack prints the current count and calls a higher stack, passing it a+1 and b. The rest of the counting up job is left to the stacks above. Once the stack above returns, the calling stack prints the returned value as part of the count down, and then returns the same value, decremented by one to the stack below to continue the count down. This form of stack programming is known as recursion, and may take careful scrutiny to fully understand. In practice, subroutines should remain very short and simple to facilitate debugging. Consider the stack a way to delegate common tasks to a small subroutine, as shown in the array zeroing ***Code 3***.

To download the latest firmware, visit http://www.galilmc.com/support/firmware-downloads.php. For more information and code examples, see JS and EI in the command reference and chapter 7 in the user manual, both attainable from http://www.galilmc.com/support/manuals.php or call Galil today to speak with an Application Engineer. ■

Figure 3: An example of the program stack running the count up/down program.



Code 4: Recursive Algorithm for counting.

```
VF2.0;'Make numbers look tidy in printouts
XQ#thread1,1;'Start thread 1 at label thread1
EN;'End thread zero
#thread1
MG"Welcome to thread ", _ED4," stack ",_ZS1
JS#count(7,9);'push nums on stack1, run code #count
MG"Welcome back to stack ", _ZS1
MG"See you later Alligator"
EN
#count;NO( ^a countFrom by value, ^b countTo by value)
 IF (^a < ^b);'If first number less than second
  MG ^a," Alligator (up, Stack", _ZS1,")";'Count first number
  JS #count(^a+1 , ^b);'push one higher number on the stack
 ELSE
  MG ^a," Alligator, now counting down (Stack",   _ZS1,")";'Done counting up
  EN,,^b-1;'return ending value - 1
 ENDIF
 MG _JS," Alligator (down, Stack", _ZS1,")";'print value returned from above
EN,,(_JS-1);'return one less to stack below
```

```
'TERMINAL OUTPUT:
 Welcome to thread 1 stack 0
 7 Alligator (up, Stack 1)
 8 Alligator (up, Stack 2)
 9 Alligator, now counting down (Stack 3)
 8 Alligator (down, Stack 2)
 7 Alligator (down, Stack 1)
 Welcome back to stack 0
 See you later Alligator
```

GALIL
WE MOVE THE WORLD

Galil. We Move the World.

With over 500,000 controllers installed worldwide, Galil is the leading supplier of motion controllers. Galil's legacy of innovation began in 1983 with the introduction of the first microprocessor-based servo motion controller. Today, Galil continues its leadership by offering the most powerful, cost-effective and easy-to-use controllers to accommodate all your motion and I/O needs.

Galil offers a broad array of motion controllers in a variety of formats: single and multi-axis, card-level and box-level, bus-based and stand-alone. Galil's Ethernet/RS232 and PCI controllers are available in an Econo version for lowest cost and Accelera version for ultra high-speed performance. Plug-in, multi-axis drives for steppers and servos save space, cost and wiring. For intelligent I/O control, the RIO Pocket PLC is compact, low-cost and packed with analog and digital I/O.

At Galil, we share our expertise with our customers. You will find a wealth of information on our website at http://www.galilmc.com. Here you can view any of Galil's free web-tutorials, read an application note or white paper, post a question on our bulletin board, or download the latest software and manuals.

Exceptional application support is a top priority at Galil. Call Galil today at 800-377-6329 to discuss your project with one of our highly-trained applications engineers.

GALIL
WE MOVE THE WORLD

ACCELERA CONTROLLERS AND DRIVES

DMC-40x0	Ethernet/RS232
DMC-18x6	PCI

ECONO CONTROLLERS AND DRIVES

DMC-21x3	Ethernet/RS232
DMC-18x2	PCI

SINGLE-AXIS CONTROLLERS AND DRIVES

DMC-1415	Ethernet/RS232
CDS-3310	Ethernet/RS232

POCKET PLC I/O CONTROLLER

RIO-47xxx	Ethernet/RS232

SOFTWARE TOOLS

GalilTools. Servo Tuning and analysis software

Ladder Interface. Converts Ladder program into DMC code for RIO Pocket PLC.

Check in daily with Galil Motion Control on LinkedIn, Twitter and Facebook!

FREE	Online Support Tools
✓	Application notes, white papers and industry articles http://www.galilmc.com/support/application-notes.php
✓	Over 20 tutorials about servo tuning, motion programming, & motors and drives http://www.galilmc.com/learning/tutorials.php
✓	MotionCode™ Toolkit, step-by-step solutions with downloadable code http://www.galilmc.com/learning/motioncode.php
✓	MotorSizer™ Tool for quick sizing of stepper and servo motion systems http://www.galilmc.com/learning/motorsizer.php
✓	Interactive bulletin board with knowledge base for fast answers to your questions http://www.galilmc.com/support/forums.php
✓	Customer stories and videos http://www.galilmc.com/support/smartmoves.php
✓	Product catalog with specs and pricing http://www.galilmc.com/support/catalog.php